An Archive Based Particle Swarm Optimisation for Feature Selection in Classification

Bing Xue
School of Engineering
and Computer Science
Victoria University of Wellington
Wellington, New Zealand
Email: Bing.Xue@ecs.vuw.ac.nz

A. K. Qin
School of Computer Science
and Information Technology
RMIT University, Melbourne,
VIC, Australia
Email: kai.qin@rmit.edu.au

Mengjie Zhang
School of Engineering
and Computer Science
Victoria University of Wellington
Wellington, New Zealand
Email: Mengjie.Zhang@ecs.vuw.ac.nz

***Abstract*—Feature selection aims to select a subset of relevant features from typically a large number of original features, which is a difficult task due to the large search space. Particle swarm optimisation (PSO) is a powerful search technique, but there are some limitations on using the standard PSO for feature selection. This paper proposes a new PSO based feature selection approach, which introduces an external archive to store promising solutions obtained during the search process. The solutions in the archive serve as potential leaders (i.e. global best, $gbest$) to guide the swarm to search for an optimal feature subset with the lowest classification error rate and a smaller number of features. The proposed approach has two specific methods, PSOArR and PSOArRWS, where PSOArR randomly selects $gbest$ from the archive and PSOArRWS uses the roulette wheel selection to select $gbest$ considering both the classification error rate and also considering the number of selected features. Experiments on twelve benchmark datasets show that both PSOArR and PSOArRWS can successfully select a smaller number of features and achieve similar or better classification performance than using all features. PSOArR and PSOArRWS outperform a PSO based algorithm without using an archive and two traditional feature selection methods. The performance of PSOArR and PSOArRWS are similar to each other.**

I. Introduction

Machine learning and data mining tasks, such as classification, often involve data having a large number of features, but irrelevant and redundant features may degrade the classification performance. Feature selection is to select a small subset of relevant features to maintain or even improve the classification performance [1]. By removing irrelevant or redundant features, feature selection can reduce the dimensionality of the dataset and thus may simplify the learnt models, speed up the model training and improve the generalization performance [1, 2].

Existing feature selection algorithms can be generally classified into two categories, filter approaches and wrapper approaches [1, 3]. Their main difference is whether a classification/learning algorithm is used during the feature selection process. A wrapper algorithm typically includes a classification algorithm to measure the classification performance of the selected features to evaluate the goodness of the selected features. Filter approaches are independent of any classification algorithm. Filter approaches are argued to be computationally efficient and more general than wrappers, but wrapper approaches can usually achieve better classification performance than filter approaches due to the direct link between the selected features and the classification algorithm. This work mainly focuses on the wrapper feature selection.

Selecting an optimal feature subset is a challenging task due to the large search space, where the number of possible solutions is 2^n for a dataset with n features [2]. Therefore, an exhaustive search is not possible in most cases and a powerful search technique is needed for developing a feature selection algorithm. Particle swarm optimisation (PSO) [4, 5] is an evolutionary computation (EC) technique, which has been successfully used in a variety of fields, including feature selection [6, 7, 8]. In PSO, the use of global best ("$gbest$") is one of the key components, but the original "$gbest$" determination method has some limitations for addressing feature selection problems (details can be seen in Section III). Therefore, to further investigate the capability of PSO for feature selection, a new $gbest$ determination method is demanded.

A. Goals

The overall goal of this paper is to develop a new PSO based feature selection approach to selecting a small subset of features to reduce the dimensionality of the dataset and achieve similar or even better classification performance than using all features. To achieve this goal, an archive is introduced to PSO to store promising solutions from which the $gbest$ of particles are chosen to improve the search ability of the PSO based algorithm. Two selection methods (i.e. random selection and roulette wheel selection) are used to select a solution from the archive as the $gbest$ of a particle. The newly developed approach is tested and compared with a PSO based algorithm without using an archive and two traditional feature selection methods on twelve datasets of varying difficulty. Specifically, we will investigate:

- whether the newly proposed approach using the random selection to choose a "$gbest$" from the archive can be used to reduce the number of features and maintain or even increase the classification accuracy over using all features,
- whether the newly proposed approach using the roulette wheel selection can select a smaller subset of relevant

features, and can achieve better performance than using the random selection method, and

- whether these new methods can outperform the PSO based method without using an archive and the two traditional feature selection methods.

B. Organisation

The remainder of the paper is organised as follows. Section 2 presents the background information of this work. Section 3 describes the new PSO based feature selection approach. Section 4 describes experimental design and Section 5 presents experimental results with discussions. Section 6 describes conclusions and future work.

II. Background

A. Particle Swarm Optimisation (PSO)

PSO [4, 5] is inspired by social behaviours such as fish schooling and bird flocking. Each solution of the target problem is represented by a particle. A swarm of particles move ("fly") together in the search space to find the best solutions. For any particle i, a vector $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$ is used to represent its position and a vector $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$ is used to represent its velocity, where D means the dimension of the target problem. During the search process, each particle can remember its best position visited so far called the personal best (denoted by *pbest*), and the best previous position visited so far by the whole swarm called global best (denoted by *gbest*). Based on personal best and global best, PSO iteratively updates x_i and v_i of each particle to search for the optimal solutions according to Eqs. 1 and 2.

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_{i1} * (p_{id} - x_{id}^{t}) + c_2 * r_{i2} * (p_{gd} - x_{id}^{t}) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2)$$

where v_{id}^{t+1} shows the velocity of particle i in the dth dimension in the $t+1$th iteration. w is the inertia weight, which indicates the influence of the previous velocity. c_1 and c_2 are acceleration constants. r_{i1}, r_{i2} and $rand()$ are random valuess, which are uniformly distributed in [0, 1]. p_{id} and p_{gd} shows the values of personal best and global best in the dth dimension. A predefined maximum velocity, v_{max}, is used to limit v_{id}^{t+1} to $[-v_{max}, v_{max}]$.

B. Related Work on Feature Selection

In this section, typical feature selection algorithms are reviewed.

1) Traditional Feature Selection Methods.: Sequential forward selection (SFS) [9] and sequential backward selection (SBS) [10] are two commonly used wrapper feature selection algorithms. Both of them use a greedy hill-climbing search strategy to search for the optimal feature subset. However, both SFS and SBS suffer from the so-called nesting effect, which means that once a feature is selected (discarded) it cannot be discarded (selected) later. Therefore, both SFS and SBS are easily trapped in local optima [2]. In addition, both SFS and SBS require long computational time when the number of features is large [2]. In order to avoid the nesting effect, Stearns [11] proposed a "plus-l-take away-r" method in which SFS is applied for l times and then SBS is applied for r backward steps. However, determining the best values of (l, r) is a challenging task.

Later, Pudil et al. [12] propose two floating selection methods, i.e. sequential backward floating selection (SBFS) and sequential forward floating selection (SFFS) to automatically determine the values of (l, r). Although the floating methods are claimed to be at least as good as the best sequential method, they are still likely to get trapped in a local optimal solution even the criterion function is monotonic and the scale of the problem is small. Meanwhile, based on the best-first algorithm and SFFS, Gutlein et al. [13] propose a linear forward selection (LFS) in which the number of features considered in each step was restricted. Experiments show that LFS improves the computational efficiency of sequential forward methods while maintaining the comparable classification accuracy of the selected feature subset.

2) EC Approaches for Feature Selection.: EC algorithms have been applied to feature selection problems, such as PSO, genetic algorithms (GAs) [14], genetic programming (GP) [15], and ant colony optimisation (ACO) [16]. Zhu et al. [14] propose a feature selection method using a memetic algorithm that is a hybridisation of local search and GA. Experiments show that this algorithm outperforms GA alone and other algorithms. Neshatian and Zhang [15] further propose a GP relevance measure (GPRM) to evaluate and rank feature subsets in binary classification tasks. Experiments show that the proposed method could detect relevant subsets of features in different situations including multi-modal class distributions and mutually correlated features, where other methods had difficulties. Based on ant colony optimisation (ACO), Kanan and Faez [16] develop a wrapper feature selection algorithm, which outperforms GA and other ACO based algorithms on a face detection dataset, but its performance has not been tested on other problems.

Chuang et al. [7] apply the so-called catfish effect to PSO for feature selection, which is to introduce new particles into the swarm by re-initialising the worst particles when $gbest$ has not improved for a number of iterations. Xue et al. [17] develop new initialisation and $pbest$ and $gbest$ updating mechanisms in PSO for feature selection, which can increase the classification accuracy and reduce both the number of features and the computational time. Lin et al. [18] propose a wrapper feature selection algorithm using PSO and support vector machine (SVM). This algorithm aims to optimise the parameters in SVM and search for the best feature subset simultaneously. Mohemmed et al. [19] propose a hybrid method (PSOAdaBoost) that incorporates PSO with an AdaBoost framework for face detection. PSOAdaBoost aims to search for the best feature subset and determine the decision thresholds of AdaBoost simultaneously, which speeds up the training process and increases the accuracies of weak classifiers in AdaBoost. Javani et al. [20] apply PSO for

Algorithm 1: Pseudo-code of the New Approach

```
begin
    split the instances into a Training and a Test set;
    initialise $x$ and $v$ of each particle;
    initialise archive ($Archive$);
    while $Maximum\ Iterations\ has\ been\ not\ met$ do
        calculate the fitness value of each particle, i.e. the
        classification error rate of the selected features;
        update Archive;
        for i=1 to $Swarm\ Size$ do
            update the personal best ($pbest$) of particle $i$;
            select a $gbest$for particle $i$ from Archive;
        end
        for i=1 to $Swarm\ Size$ do
            for d=1 to $Dimensionality$ do
                calculate $v_i$ according to Equation (1)
                calculate $x_i$ according to Equation (2)
            end
        end
    end
    calculate the classification performance of the selected
    features on the test set;
    return solutions in the $Archive$ and theirs training and
    testing classification performances;
end
```

feature selection and clustering, where each particle is used to optimise the weights of all features and the cluster center values. Feature selection is achieved by removing features with low weights. However, features with low weights may be useful because of feature interaction and the removal of them may reduce the performance. More recent work on EC for feature selection can be seen in [21, 22, 23, 24, 25, 26]

III. Proposed Approach

In feature selection, when using the classification performance as the fitness function, feature subsets, which have different numbers of features and different combinations of individual features, may have the same classification accuracy. Therefore, there may exist multiple distinct feature subsets sharing the lowest classification error rate. All of these feature subsets should have a chance to become $gbest$ and the one with a smaller number of features should have a larger probability to be selected as $gbest$. However, this cannot be achieved by the original way of updating $gbest$. Therefore, we develop a new PSO based approach by introducing an archive to store promising solutions from which $gbest$ is chosen. With this archive, all feature subsets with the low classification error rate and a smaller number of features have the chance to be selected as $gbest$ to guide the search of the algorithm, which can also help prevent the swarm from quickly losing its diversity.

A. Overview of the New Approach

Algorithm 1 shows the pseudo-code and the overall structure of the new approach. The new approach follows the main steps of a standard PSO algorithm except for the use of an archive and the update/selection of $gbest$, which are described in details as follows.

B. Update Archive

During the search process, new and better solutions (feature subsets) will be obtained by the PSO algorithm. Therefore, the archive needs to be updated by adding a newly found solution (s) and at the same time removing the solution (s) with the worst classification performance and the largest number of features.

A newly found solution is added into the archive in two situations. The first situation is when the newly found solution can achieve better classification performance than any solution in the archive. The second situation is that when the newly found solution can achieve at least the same classification performance as any solution in the achieve, but includes a smaller number of features. In this way, the archive will maintain the feature subsets with the lower classification error rates and a smaller numbers of features.

C. Selection of gbest

The archive is introduced to store potential $gbest$ to lead the algorithm to search for feature subsets with the highest classification accuracy. Two $gbest$ selection methods are used to select a $gbest$ from the archive for each particle. The first one is random selection, where $gbest$ is randomly selected from the archive. The second method is using roulette wheel selection to choose $gbest$ from the archive according to the classification performance and then the number of features. The solutions with better classification performance have larger probabilities to be selected. If multiple solutions have the same classification performance but different numbers of features, the one with the smallest number of features has the largest probability to be selected. By considering different probabilities, we expect the roulette wheel selection will select a better $gbest$ to guide the algorithm to find a better feature subset than using the random selection.

The new algorithm using the random selection is named PSOArR and the algorithm using the roulette wheel selection is named PSOArRWS in this paper. Note that the $pbest$ is updated purely according to the fitness function and independent of the archive.

D. Fitness Function

Eq. 3, which measures the classification error rate, is used as the fitness function in both PSOArR and PSOArRWS. By minimising Eq. 3, PSOArR and PSOArRWS aims to minimise the classification error rate (or maximise the classification accuracy) of the selected features.

$$Fitness = ErrorRate = \frac{FP + FN}{TP + TN + FP + FN} \tag{3}$$

where FP, FN, TP and TN stand for false positives, false negatives, true positives, and true negatives, respectively.

E. Representation

In this paper, continuous PSO is used and the binary version of PSO [27] is not used because the current binary PSO has some limitations [28]. Therefore, in PSOArR and PSOArRWS, each particle is represented by a vector of real numbers, $x_i = (x_{i1}, x_{i2}, .., x_{id}, .., x_{iD})$, where D is the dimensionality or the total number of features in the dataset. $0 \leq x_{id} \leq 1$ shows the probability of the dth feature being selected. A threshold θ is used to determine whether this feature is selected. If $\theta \leq x_{id}$, the dth feature is selected. Otherwise, the dth feature is not selected.

Note that the external archive is often used in multi-objective algorithms [29], but PSOArR and PSOArRWS are single objective algorithms. The main reason is that they do not intend to find a set of trade-off solutions, but to find a solution with the best classification performance and the smallest number of features. The archive is used to lead the swarm searching different regions of the solution space and avoid the stagnation in local optima. The objective of minimising the number of features is implicitly considered in PSOArR and PSOArRWS. When selecting $gbest$, the number of features is also considered, which helps reduce the number of features.

IV. Design of Experiments

A. Benchmark Techniques

To examine the performance of the proposed algorithms (PSOArR and PSOArRWS), two traditional wrapper feature selection methods and a PSO based feature selection algorithm (PSOFS) [8] without using the archive are used as benchmark techniques in the experiments.

The two traditional algorithms are linear forward selection (LFS) [13] and greedy stepwise backward selection (GSBS), which were derived from two typical greedy search based feature selection, i.e. SFS and SBS, respectively. LFS restricts the number of features to be considered in each step of the forward selection, which can reduce the number of evaluations. Therefore, LFS is computationally less expensive and can usually obtain better results than SFS. More details can be seen from the literature [13]. GSBS starts with all available features and stops when the deletion of any remaining feature results in a decrease in the classification performance. PSOFS uses the classification error rate as the fitness function, which is the same as PSOArR and PSOArRWS. PSOFS does not involve any archive and standard PSO is used to search for the feature subset having the lowest classification error rate.

B. Datasets and Parameter Settings

Twelve datasets (Table I) chosen from the UCI machine learning repository [30] are used in the experiments. The twelve datasets are chosen to have different numbers of features, classes and instances to be used as the representatives of problems that the proposed algorithms can address. For each dataset, the instances are randomly divided into two sets: 70% as the training set and 30% as the test set.

In the experiments, K-nearest neighbour (KNN), where K=5, is used as the classification/learning algorithm. During the training process, KNN with 10-fold cross-validation is employed to evaluate the classification error rate of the selected feature subset on the training set, and then the selected features are evaluated on the test set to calculate the testing classification error rate. A detailed discussion of why and how 10-fold cross-validation is applied in this way is given by [2].

TABLE I
Datasets

Dataset	# Features	# Classes	# Instances
Wine	13	3	178
Zoo	17	7	101
Vehicle	18	4	846
German	24	2	1000
Ionosphere	34	2	351
Sonar	60	2	208
Hill-Valley	100	2	606
Musk Version 1 (Musk1)	166	2	476
Semeion	256	2	1593
Madelon	500	2	4400
Isolet5	617	2	1559
Multiple Features	649	10	2000

Weka [31] is used to run the experiments of LFS and GSBS. All the settings in LFS and GSBS are kept to the defaults. The parameters of PSOArR, PSOArRWS and PSOFS are set as follows: $w = 0.7298$, $c_1 = c_2 = 1.49618$, $v_{max} = 6.0$, the population size is 30, and the maximum iteration is 100. The fully connected topology is used. These values are chosen based on the common settings in [5]. PSOFS shares the same representation as PSOArR and PSOArRWS. The threshold θ is set as 0.6 in the experiments to be slightly biased on selecting a smaller number of features. For each dataset, each algorithm has been executed for 50 independent runs.

V. Results and Discussions

Since PSOArR and PSOArRWS were designed as single objective algorithms, they reports a single solution from each run. The archive in both PSOArR and PSOArRWS contains multiple solutions (feature subsets), where the solution with the highest classification accuracy (if multiple solutions share the highest classification accuracy, the one with the smallest number of features) is selected as the final solution. Fig. 1 presents the results of PSOArR, PSOArRWS and PSOFS, where 50 results (one from each of the 50 runs) are plotted for each algorithm in each chart. To further investigate the potential ability of the three algorithms, Figs. 2 compares the non-dominated solutions achieved by each algorithm in the 50 independent runs, which is to show the best solutions that can be achieved by each algorithm.

In Figs. 1 and 2, each chat corresponds one of the datasets used in the experiments. On the top of each chart, the numbers in the bracket show the total number of features and the error rate obtained by using all features for classification. It is noted that in Fig. 1, although 50 solutions are plotted, less than 50 distinguished points are shown. The main reason is that many solutions may have the same classification performance and the same number of features. So they are plotted as the same point, although they may select different individual features.

A. Results of PSOArR and PSOArRWS

According to Figs. 1 and 2, it can be seen that in *all* the twelve datasets, both PSOArR and PSOArRWS selected

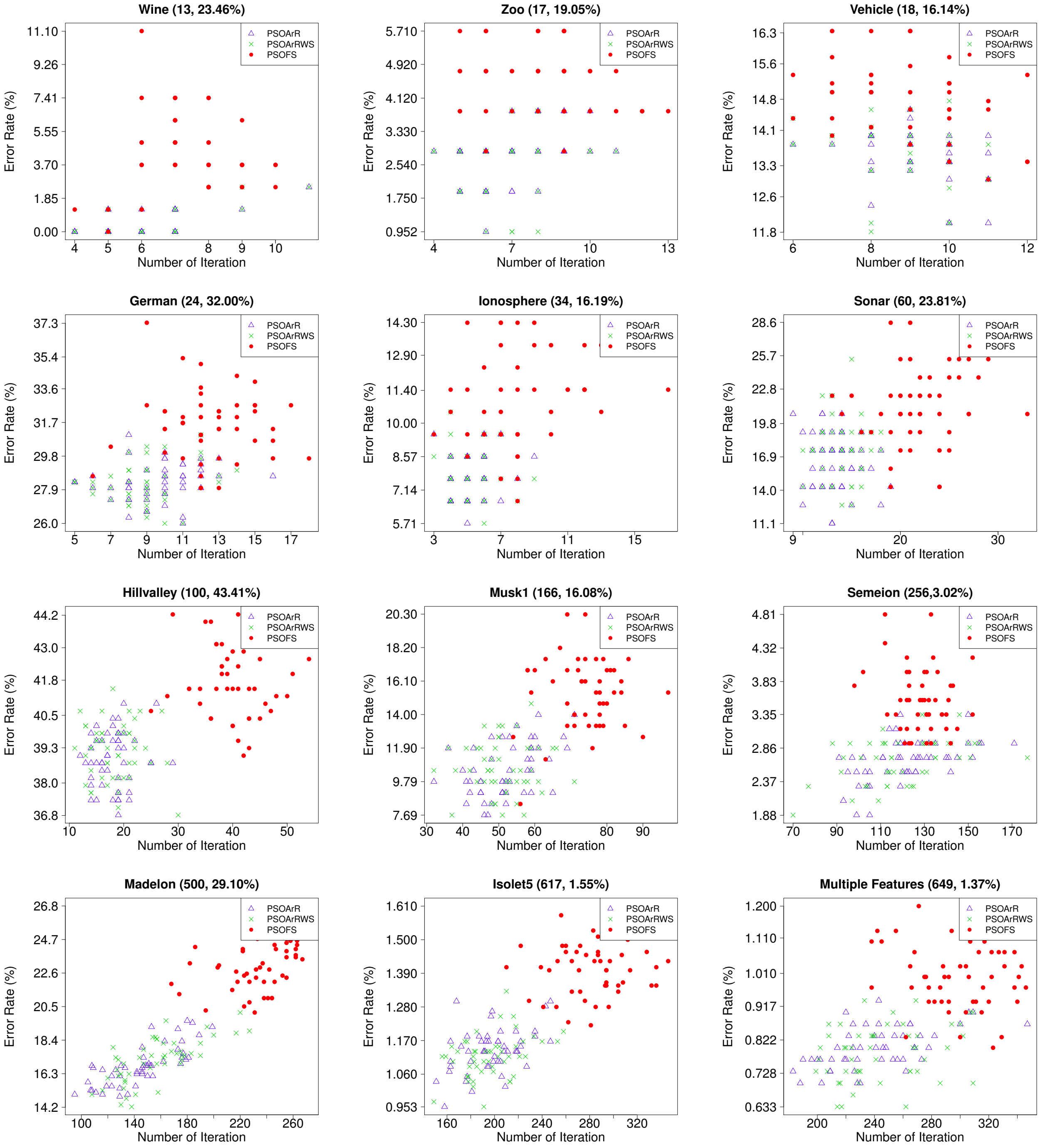

Fig. 1. Experimental Results of PSOArR, PSOArRWS and PSOFS

a smaller number of features and achieved the similar or even lower classification error rate than using all features. In almost *all* cases, both PSOArR and PSOArRWS include a feature subset, which selects less than 30% of the available features and achieved better classification performance than using all features. For example, as can be seen from Fig. 2, on the Ionoshpere dataset, PSOArR and PSOArRWS produced feature subsets including only two of the 34 features and reducing the classification error rate from 16.19% to 10.5% in PSOArRWS and to 9.5% in PSOArR.

The results suggest that both PSOArR and PSOArRWS can be successfully used for feature selection to reduce the dimensionality of the data and maintain or improve the classification performance.

From Figs. 1 and 2, it can also be seen that in most cases, there is not much difference between PSOArR and PSOArRWS although they use different *gbest* selection methods. This is different from our original hypothesis, where we expected

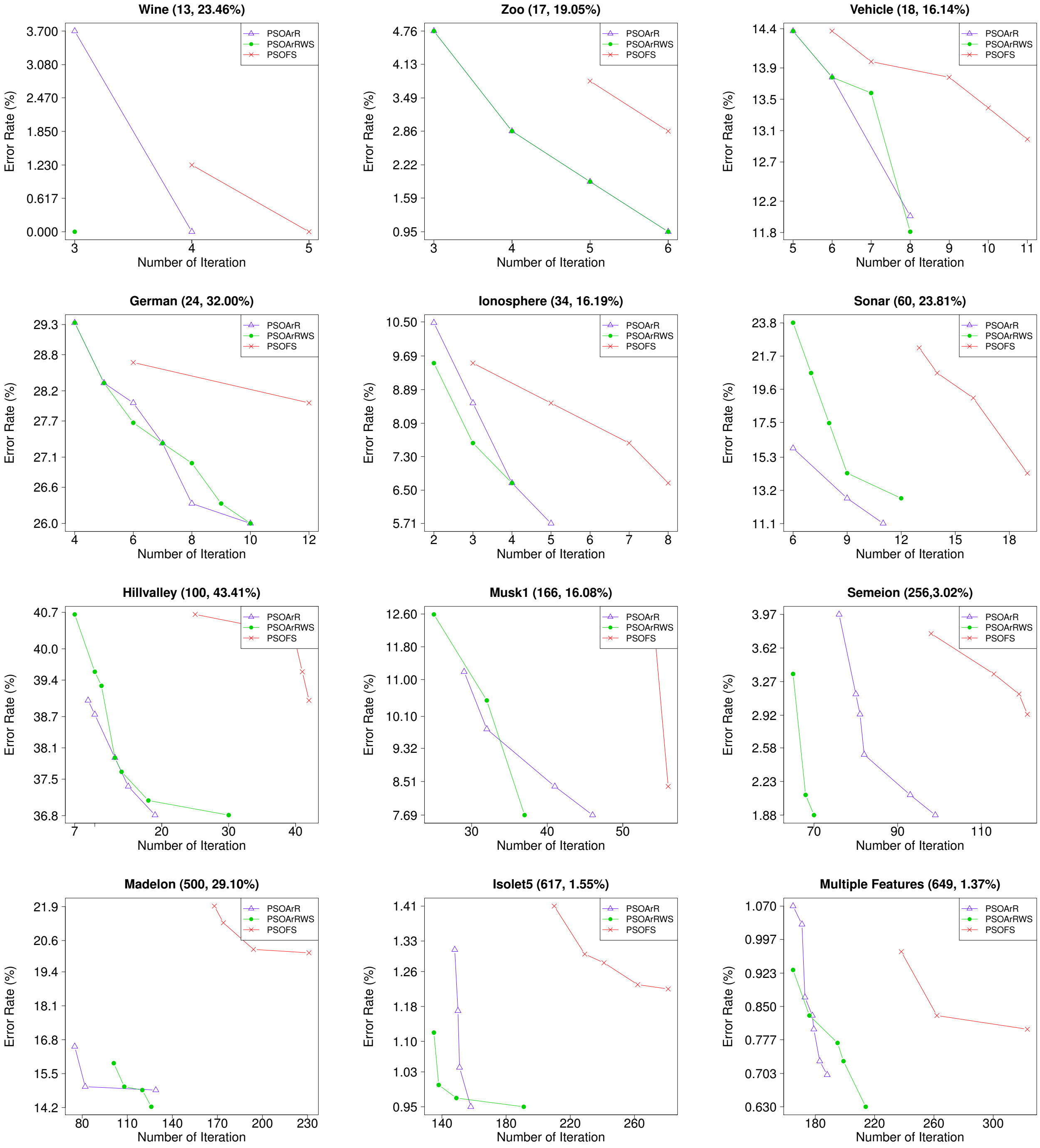

Fig. 2. Experimental Results of PSOArR, PSOArRWS and PSOFS

that PSOArRWS can achieve better results than PSOArR. A detailed check reveals that the solutions in the archive have the similar classification error rate and the similar number of features. The *gbest* selected by PSOArR and PSOArRWS are similar to each other in almost all cases.

B. Comparisons with PSOFS

1) Comparisons on overall performance: According to Fig. 1, in all cases, the classification error rate of PSOFS is larger than that of PSOArR and PSOArRWS. On almost all datasets, the number of features selected by PSOArR and PSOArRWS is smaller or much smaller than that of PSOFS, especially on the datasets with a relatively large number of features. For example, on the datasets with more than 500 features (i.e. Madelon, Isolet5 and Multiple Features), PSOArR and PSOArRWS achieved a lower classification error rate and selected a much smaller number of features than PSOFS.

The results suggest that although PSOArR, PSOArRWS and

TABLE II
RESULTS OF LFS ANG GSBS

	Wine		Zoo		Vehicle		German		Ionosphere		Sonar	
	Size	Error	Size	Error	Size	Error	Size	Error	Size	Error	Size	Error
LFS	7	25.93	8	20.95	9	16.93	3	31.33	4	13.33	3	22.22
GSBS	8	14.81	7	20	16	24.21	18	35.67	30	21.9	48	31.75

	Hillvalley		Musk1		Semeion		Madelon		Isolet5		Multi.Fea.	
	Size	Error	Size	Error	Size	Error	Size	Error	Size	Error	Size	Error
LFS	8	42.31	10	14.69			7	35.38	24	1.66	18	1.0
GSBS	90	50.55	122	23.78			489	48.72	560	2.84		

PSOFS shares the same fitness function, the use of the archive in PSOArR and PSOArRWS help further reduce the number of features and improve the classification performance.

2) Comparisons on non-dominated solutions: As can be seen from Fig. 2, it is obvious that the non-dominated solutions of PSOArR and PSOArRWS achieved better performance than PSOFS in terms of both the classification performance and the number of features in *all* the twelve datasets. On the datasets with more than 500 features, the number of features selected in PSOArR and PSOArRWS is only around half of that in PSOFS, but the classification performance is still better.

The results further show that PSOArR and PSOArRWS using the archive can help them reduce the number of features and improve the classification performance over PSOFS.

C. Comparisons with Traditional Methods

Table II shows the results of the two traditional feature selection algorithms, where the results of the Semeion and Multiple Features datasets are not available because the search process can not be finished within two days or the algorithm in Weka fails to produce any solution. Both LFS and GSBS are deterministic algorithms that produce a unique solution.

Comparing the results in Table II to that in Fig. 2, it can be seen that in *all* datasets, PSOArR and PSOArRWS selected smaller numbers of features, but achieved lower classification error rates than LFS and GSBS, especially for GSBS on datasets with a large number of features.

D. Further Comparisons with Multi-objective Approaches

Fig. 2 shows that PSOArR and PSOArRWS outperforms the PSO based single objective algorithm in terms of both the classification performance and the number of features. In order to further test their performance, we compare the results of PSOArR and PSOArRWS with a PSO based multi-objective feature selection approach (CMDPSOFS) [8] because PSOArR and PSOArRWS produce multiple solutions in a single run, which is similar to that of the multi-objective algorithm CMDPSOFS.

In [8], CMDPSOFS was proposed to simultaneously minimise the number of features and maximise the classification accuracy (i.e. minimise the classification error rate) [8]. Noted that it is not entirely fair to compare PSOArR and PSOArRWS with CMDPSOFS, but we compare their results to further test whether PSOArR and PSOArRWS as single objective algorithms can achieve better classification performance than the multi-objective algorithm. The detailed results of CMDPSOFS are not presented here due to the page limit. Comparing Fig. 2 with the results in [8], it is observed that in most cases, the classification performance of PSOArR and PSOArRWS is slightly better than CMDPSOFS. However, PSOArR and PSOArRWS selected a similar number of features to CMDPSOFS on datasets with a small number of features, but they selected a larger number of features on high dimensional datasets. This is clearly because CMDPSOFS employs a multi-objective search mechanism while PSOArR and PSOArRWS follow a single objective search mechanism. From the classification accuracy point of view, PSOArR and PSOArRWS have their advantage over CMDPSOFS. Therefore, if users has high demand on the classification performance, PSOArR or PSOArRWS is a better choice than CMDPSOFS, but if users require a set of trade-off solutions or have high demand on reducing the dimensionality, CMDPSOFS is preferred over PSOArR or PSOArRWS.

Further reducing the number of features without reducing or even increasing the classification performance needs to be done in our future work.

VI. CONCLUSIONS AND FUTURE WORK

The goal of this work was to develop a new PSO based feature selection approach to reduce the dimensionality of the data and maintain or even improve the classification performance over using all features. This goal has been successfully achieved by introducing an external archive to PSO for feature selection, which stores the potential *gbest* solutions. Two new algorithms, PSOArR and PSOArRWS, were then developed by using random selection and roulette wheel selection to choose *gbest* from the archive, respectively. The two new algorithms were examined and compared with a PSO based algorithm without using an archive and two traditional feature selection methods. The results show that both PSOArR and PSOArRWS can successfully address feature selection problems to reduce the dimensionality of the data and improve the classification performance. In most cases, they outperformed the other three methods in terms of the classification performance and the number of features. PSOArR and PSOArRWS achieved the similar performance to each other because the solutions in the archive have a similar classification error rate and a similar number of features and the *gbest* selection method does not produce significantly different results.

In the future, we will further investigate the potential of PSO and other evolutionary computation techniques for feature selection on datasets with a larger number of features, i.e. large-scale problems. We also intend to develop multi-objective feature selection algorithms to simultaneously maximise the classification accuracy and minimise the number of features.

REFERENCES

[1] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 4, pp. 131–156, 1997.

[2] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, pp. 273–324, 1997.

[3] L. Y. Chuang, H. W. Chang, C. J. Tu, and C. H. Yang, “Improved binary PSO for feature selection using gene expression data,” *Computational Biology and Chemistry*, vol. 32, no. 29, pp. 29– 38, 2008.

[4] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[5] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *IEEE International Conference on Evolutionary Computation (CEC’98)*, 1998, pp. 69–73.

[6] B. Xue, M. Zhang, and W. N. Browne, “New fitness functions in binary particle swarm optimisation for feature selection,” in *IEEE Congress on Evolutionary Computation (CEC’12)*, 2012, pp. 2145–2152.

[7] L. Y. Chuang, S. W. Tsai, and C. H. Yang, “Improved binary particle swarm optimization using catfish effect for feature selection,” *Expert Systems with Applications*, vol. 38, pp. 12 699–12 707, 2011.

[8] B. Xue, M. Zhang, and W. Browne, “Particle swarm optimization for feature selection in classification: A multi-objective approach,” *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2013.

[9] A. Whitney, “A direct method of nonparametric measurement selection,” *IEEE Transactions on Computers*, vol. C-20, no. 9, pp. 1100–1103, 1971.

[10] T. Marill and D. Green, “On the effectiveness of receptors in recognition systems,” *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 11–17, 1963.

[11] S. Stearns, “On selecting features for pattern classifier,” in *Proceedings of the 3rd International Conference on Pattern Recognition*. 1976, pp. 71–75.

[12] P. Pudil, J. Novovicova, and J. V. Kittler, “Floating search methods in feature selection,” *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[13] M. Gutlein, E. Frank, M. Hall, and A. Karwath, “Large-scale attribute selection using wrappers,” in *IEEE Symposium on Computational Intelligence and Data Mining (CIDM ’09)*. IEEE, 2009, pp. 332–339.

[14] Z. X. Zhu, Y. S. Ong, and M. Dash, “Wrapper-filter feature selection algorithm using a memetic framework,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 1, pp. 70–76, 2007.

[15] K. Neshatian and M. Zhang, “Genetic programming for feature subset ranking in binary classification problems,” in *European Conference on GP*.2009, pp. 121–132.

[16] H. R. Kanan and K. Faez, “An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system,” *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 716–725, 2008.

[17] B. Xue, M. Zhang, and W. Browne, “Novel initialisation and updating mechanisms in PSO for feature selection in classification,” in *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science. 2013, vol. 7835, pp. 428–438.

[18] S. W. Lin, K. C. Ying, S. C. Chen, and Z. J. Lee, “Particle swarm optimization for parameter determination and feature selection of support vector machines,” *Expert Systems with Applications*, vol. 35, no. 4, pp. 1817–1824, 2008.

[19] A. Mohemmed, M. Zhang, and M. Johnston, “Particle swarm optimization based adaboost for face detection,” in *IEEE Congress on Evolutionary Computation (CEC’09)*, 2009, pp. 2494–2501.

[20] M. Javani, K. Faez, and D. Aghlmandi, “Clustering and feature selection via pso algorithm,” in *International Symposium on Artificial Intelligence and Signal Processing (AISP’11)*, 2011, pp. 71 –76.

[21] B. Xue, L. Cervante, L. Shang, W. N. Browne, and M. Zhang, “A multi-objective particle swarm optimisation for filter based feature selection in classification problems,” *Connection Science*, 2012.

[22] B. Xue, M. Zhang, and W. N. Browne, “Multi-objective particle swarm optimisation (pso) for feature selection,” in *Genetic and Evolutionary Computation Conference (GECCO’12)*. ACM, 2012, pp. 81–88.

[23] L. Cervante, B. Xue, M. Zhang, and L. Shang, “Binary particle swarm optimisation for feature selection: A filter based approach,” in *IEEE Congress on Evolutionary Computation (CEC’12)*, 2012, pp. 881–888.

[24] K. Neshatian, M. Zhang, and P. Andreae, “A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 645–661, 2012.

[25] B. Xue, L. Cervante, L. Shang, W. N. Browne, and M. Zhang, “Multi-objective evolutionary algorithms for filter based feature selection in classification,” *International Journal on Artificial Intelligence Tools*, vol. 22, no. 04, p. 1350024, 2013.

[26] M. Lane, B. Xue, I. Liu, and M. Zhang, “Particle swarm optimisation and statistical clustering for feature selection,” in *AI 2013: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science. 2013, vol. 8272, pp. 214–220.

[27] J. Kennedy and R. Eberhart, “A discrete binary version of the particle swarm algorithm,” in *IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation.*, vol. 5, 1997, pp. 4104–4108.

[28] A. P. Engelbrecht, *Computational intelligence: an introduction (2. ed.)*. Wiley, 2007.

[29] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK: John Wiley & Sons, 2001.

[30] A. Frank and A. Asuncion, “UCI machine learning repository,” 2010.

[31] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, “The WEKA Data Mining Software: An Update” *SIGKDD Explorations*, vol. 11, 2009.